TRANSIMS Data Preparation Guide

Revision 2.1

B. W. Bush

Energy and Environmental Analysis Group
Los Alamos National Laboratory

K. P. Berkbigler

Computer Research and Applications Group
Los Alamos National Laboratory

L. L. Smith

Technology and Safety Assessment Division
Los Alamos National Laboratory

8 April 1998

Contents

I. Introduction

The TRANSIMS (TRansportation Analysis SIMulation System) software package requires a variety of network, transit, vehicle, population, activity, and land use data. This document provides an overview of the TRANSIMS data requirements and some guidance related to the collection, preparation, generation, and validation of these data.

This document is a work-in-progress. The present version focuses exclusively on road network and population data. Future revisions will include discussions of

- traveler activity information,
- transit routes and schedules, and
- vehicles and vehicle fleet mixes.

II. Road Network Data

The TRANSIMS network representation provides detailed information about streets, intersections, signals, and transit in a road network. In this section we discuss the concepts involved in describing a road network, TRANSIMS data table formats, and algorithms for automatically generating missing data values to complete the tables. In our analysis of road networks, we have relied on traffic engineering practice as described in references [Do 97], [GHA 88], [ITE 85], [ITE], [MM 84], [Or 93], and [PP 93].

A. Concepts

1. Node

A node is the part of the network corresponding to a vertex in graph theory. Nodes typically occur at intersections in the road network. A node must be present where the network branches and where the permanent number of lanes changes. A lane is considered permanent if it is not a temporary, pocket lane (see the definition of pocket lane below). A node may be present where neither of the aforementioned occurs, however. Nodes are not required where turn pockets start or end, as these are not considered permanent lanes. Each node has a traffic control associated with it (null, unsignalized, pre-timed, actuated, coordinated, etc.).

2. Link

A link is the part of the network corresponding to an edge in graph theory. Links represent street and road segments. Each link has a constant number of permanent lanes, but may have a variable number of pocket lanes. A link may have lanes in both directions; alternately, the lanes in opposite directions may be on separate links (in which case no passing into oncoming lanes is possible).

3. Functional Classes

Table 1 below lists the functional classes for links.

Table 1. Functional classes for links [Do 97].

Name	Interpretation
Freeway	A divided, arterial highway for through traffic with full control of access. Full access control means the authority to control access is exercised to give preference to through traffic by providing access connections with selected public roads, but prohibiting grade crossings and/or direct private driveway connections.
Expressway	A divided, arterial highway for through traffic with partial control of access. Partial control of access means that some authority is exercised to control access in the manner described above but there are crossings at grade and/or direct private driveway connections.
Primary Arterial	A major arterial roadway with intersections at grade crossings and direct access to abutting property and on which geometric design and traffic-control measures are used to expedite safe movement of through traffic.
Secondary Arterial	A minor arterial roadway with intersections at grade crossings and direct access to abutting property and on which geometric design and traffic-control measures are used to expedite safe movement of through traffic.
Frontage Road	An arterial that runs parallel to a freeway or expressway.
Collector Street	A roadway on which vehicular traffic is given preferential right of way, and at the entrances to which vehicular traffic from intersecting roadways is required by law to yield right-of-way to vehicles on such a roadway in obedience to either a stop sign or a yield sign, when such signs are erected.
Local Street	A street or road primarily for access to residence, business, or other abutting property.
Freeway Ramp	A unidirectional roadway providing connection between a freeway or expressway and an arterial.
Zonal Connector	An imaginary (non-physical) connection to or from the centroid of a traffic analysis zone.
Other	Any roadway not fitting the above definitions.
Walkway	A street restricted to use by pedestrians.
Busway	A street restricted to use by buses.
Light Rail	A roadbed restricted to use by light rail cars.
Heavy Rail	A roadbed restricted to use by heavy rail cars.
Ferry	A waterway crossed by ferry.

4. Lane

A lane is where traffic flows on a link. The lanes on each side/direction of the link are numbered separately, starting with lane number one as the leftmost lane (relative to the direction of travel). Each successive lane to the right of it is numbered one greater than its predecessor. Pocket lanes (i.e., turn pockets, merges, and pull-outs) are numbered in sequence, even if they do not exist for the full length of the link. A two-way left-turn lane, if present, is considered to be lane number zero.

5. Pocket Lane

A pocket lane is either (a) a right- or left-turn pocket (a lane that starts after the “from” node and ends at the “to” node), (b) a right or left pull-out (a lane that starts after the “from” node and ends before the “to” node), or (c) a right or left merge pocket (a lane that starts at the “from” node and ends before the “to” node). If a lane starts at the “from” node and ends at the “to” node, it is considered a permanent lane, not a pocket lane.

6. Barrier

A barrier is a divider such as a curb or grade separation that prevents vehicles from moving between two adjacent lanes on a link.

7. Parking

Parking areas are located along links and are used as origins and destinations for vehicle trips. Parking may be placed where it is physically located in the network, or it may be placed in aggregate generic parking areas representing several of the driveways, lots, parking places, etc., on a link. Places where vehicles leave the network are called boundary parking areas.

8. Transit Stop

A transit stop is a location on a link where a transit vehicle, such as a bus or light rail car, waits to embark and disembark passengers.

9. Lane Connectivity

Lane connectivity specifies how lanes are connected across a node. Lanes are numbered from the median and include turn pockets. Incoming and outgoing links and lanes are defined relative to the node. For each incoming lane on an incoming link, at least one outgoing lane must be specified for each outgoing link that a vehicle on the incoming link can transition to. Multiple outgoing lanes may be defined for an outgoing link, if desired.

10. Traffic Control

Each node has a traffic control associated with it. The traffic control specifies how lanes are connected across the node and the type of sign or signalized control that determines who has the right-of-way.

11. Signal Coordinator

A signal coordinator is a device that controls the operation of one or more traffic controls.

12. Unsignalized Node

An unsignalized node represents the type of sign control, if any, that is present at an unsignalized node. Examples are stop and yield signs. Nodes where only the number of permanent lanes is changing are generally considered unsignalized.

13. Signalized Node

A signalized node represents a traffic light. Each signal has a timing plan and a phasing plan.

14. Phasing Plan

A phasing plan specifies the turn protection in effect for transitioning from an incoming link to an outgoing link during a particular phase of a specific timing plan.

15. Timing Plan

A timing plan specifies the lengths of the intervals during the specific phases for a traffic light. Many nodes may have the same timing plan. It is possible for each phase to transition to more than one phase if required.

16. Detector

A detector is a device that identifies the presence or passage of a vehicle over an area of the lanes on a link.

17. Study Areas and Buffer Areas

The microsimulation distinguishes two types of links in its calculations: Study area links are the links of interest for the traffic analyst. The output subsystem, for instance, records events such as when a vehicle leaves or enters the study area. The nature of the microsimulation makes it necessary also to simulate traffic on additional buffer area links. Typically, these links form a fringe about two links thick around the study area. A simulation includes buffer links in order to avoid edge effects such as when vehicles enter the study area on its boundary; the buffer gives these vehicles time to interact with other traffic and achieve realistic behavior before entering the study area.

B. Data Table Formats

In this section we specify the formats for the seventeen data tables required to describe a TRANSIMS road network. Table 2 below shows how the tables depend on one another. The units of measurement are SI units—i.e., distances in meters, time in seconds, etc. Geographic coordinates are specified in the UTM system. The TRANSIMS software architecture allows for the inclusion of additional columns desired by an analyst, so the specification below gives only the required columns. The preferred format for data files is ASCII, with columns delimited by commas, and with strings surrounded by single quotes (a single quote within a string is represented as a pair of single quotes). Records are terminated by a new-line character (i.e., ISO format). Other formats such as dBASE III and HDF are also acceptable.

Table 2. Interdependencies between network tables.

Table	Tables on which it depends
Link	Node
Speed	Node, Link, Pocket Lane
Pocket Lane	Node, Link
Lane Use	Node, Link, Pocket Lane
Parking	Node, Link
Barrier	Node, Link, Pocket Lane
Transit Stop	Node, Link
Lane Connectivity	Node, Link, Pocket Lane
Turn Prohibition	Node, Link, Pocket Lane
Unsignalized Node	Node, Link, Pocket Lane
Signalized Node	Node, Timing Plan
Phasing Plan	Node, Link, Pocket Lane, Timing Plan
Detector	Node, Link, Pocket Lane
Signal Coordinator	Node, Signalized Node
Study Area Link	Link

1. Node Table

Table 3 specifies the format for the node table. To validate a node table, it is necessary to verify the following:

- The field names and types are correct.
- The data values are in the legal ranges.
- The IDs are unique.

Table 3. Node table format.

Column Name	Description	Allowed Values
ID	ID number of the node	integer: 1 through 2,147,483,647
EASTING	*x*-coordinate of the node (in meters, UTM coordinate system)	floating-point number
NORTHING	*y*-coordinate of the node (in meters, UTM coordinate system)	floating-point number
ELEVATION	*z*-coordinate of the node (in meters, UTM coordinate system)	floating-point number
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

2. Link Table

Table 4 specifies the format for the link table. To validate a link table, it is necessary to verify the following:

- The field names and types are correct.
- The data values are in the legal ranges.
- The IDs are unique.
- The nodes at the endpoints exist.
- There are permanent lanes in at least one direction.
- There is at least one permanent lane in every direction that there is a pocket lane.
- The length of the link is at least as great as the distance between its endpoints.
- The sum of the setback lengths is less than the length of the link.
- All nodes have at least one incoming and one outgoing link.
- At least some types of vehicles are allowed on the link.
- The network graph is fully connected (i.e., one can reach any node from any other node).

Table 4. Link table format.

Column Name	Description	Allowed Values
ID	ID number of the link	integer: 1 through 2,147,483,647
NAME	name of street	50 characters
NODEA	ID number of the node at A	integer: 1 through 2,147,483,647
NODEB	ID number of the node at B	integer: 1 through 2,147,483,647
PERMLANESA	number of lanes on the link heading toward node A, not including pocket lanes	integer: 0 through 255
PERMLANESB	number of lanes on the link heading toward node B, not including pocket lanes	integer: 0 through 255
LEFTPCKTSA	number of pocket lanes to the left of the permanent lanes heading toward node A	integer: 0 through 255
LEFTPCKTSB	number of pocket lanes to the left of the permanent lanes heading toward node B	integer: 0 through 255
RGHTPCKTSA	number of pocket lanes to the right of the permanent lanes heading toward node A	integer: 0 through 255
RGHTPCKTSB	number of pocket lanes to the right of the permanent lanes heading toward node B	integer: 0 through 255

Table 4 continued.

Column Name	Description	Allowed Values
TWOWAYTURN	whether there is a two-way left-turn lane in the center of the link	one character: • 'F' = false/no • 'T' = true/yes
LENGTH	length of the link (in meters)	positive floating-point number
GRADE	percentage grade from node A to node B, uphill being a positive number	floating-point number between –100 and +100
SETBACKA	set-back distance (in meters) from the center of the intersection at node A	non-negative floating-point number
SETBACKB	set-back distance (in meters) from the center of the intersection at node B	non-negative floating-point number
CAPACITYA	total capacity (in vehicles per hour) for the lanes traveling to node A	non-negative floating-point number
CAPACITYB	total capacity (in vehicles per hour) for the lanes traveling to node B	non-negative floating-point number
SPEEDLMTA	default speed limit (in meters per second) for vehicles traveling toward node A	positive floating-point number
SPEEDLMTB	default speed limit (in meters per second) for vehicles traveling toward node B	positive floating-point number
FREESPDA	default free-flow speed (in meters per second) for vehicles traveling toward node A	positive floating-point number
FREESPDB	default free-flow speed (in meters per second) for vehicles traveling toward node B	positive floating-point number

Table 4 continued.

Column Name	Description	Allowed Values
FUNCTCLASS	functional class of the link; a link that permits both road and rail traffic should be coded with the roadway class	ten characters: • 'FREEWAY' = freeway • 'XPRESSWAY' = expressway • 'PRIARTER' = primary arterial • 'SECARTER' = secondary arterial • 'FRONTAGE' = frontage road • 'COLLECTOR' = collector • 'LOCAL' = local street • 'RAMP' = freeway ramp • 'ZONECONN' = zonal connector • 'OTHER' = other • 'WALKWAY' = walk only • 'BIKEWAY' = bicycle only • 'BUSWAY' = bus only roadway • 'LIGHTRAIL' = light rail only • 'HEAVYRAIL' = heavy rail • 'FERRY' = ferry
THRUA	default through link connected at node A; a zero indicates there is no through link	integer: 0 through 2,147,483,647
THRUB	default through link connected at node B; a zero indicates there is no through link	integer: 0 through 2,147,483,647
COLOR	the color number for the link (all of the links connected to a given link must have different colors)	integer: 1 through 63

Table 4 continued.

Column Name	Description	Allowed Values
VEHICLE	vehicle types (modes) allowed to use this link	string of characters separated by slashes: • ‘WALK’ = walking allowed • ‘AUTO’ = private auto • ‘TRUCK’ = motor carrier • ‘BICYCLE’ = bicycle • ‘TAXI’ = paratransit • ‘BUS’ = bus • ‘TROLLEY’ = trolley • ‘STREETCAR’ = streetcar • ‘LIGHTRAIL’ = light rail transit • ‘RAPIDRAIL’ = rail rapid transit • ‘REGIONRAIL’ = regional rail
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

3. Speed Table

Table 5 specifies the format for the speed table.

Table 5. Speed table format.

Column Name	Description	Allowed Values
LINK	ID number of the link with multiple speeds	integer: 1 through 2,147,483,647
NODE	ID number of the node toward which lanes are headed	integer: 1 through 2,147,483,647
SPEEDLMT	speed limit (in meters per second) for vehicles	positive floating-point number
FREESPD	free-flow speed (in meters per second) for vehicles	positive floating-point number
VEHICLE	vehicle type(s) to which speeds apply	string of characters separated by slashes: • ‘AUTO’ = private auto • ‘TRUCK’ = motor carrier • ‘BICYCLE’ = bicycle • ‘TAXI’ = paratransit • ‘BUS’ = bus • ‘TROLLEY’ = trolley • ‘STREETCAR’ = streetcar • ‘LIGHTRAIL = light rail transit • ‘RAPIDRAIL’ = rail rapid transit • ‘REGIONRAIL’ = regional rail
STARTTIME	starting time for the speeds	a character string with the day of week, • ‘SUN’ = Sunday • ‘MON’ = Monday • ‘TUE’ = Tuesday • ‘WED’ = Wednesday • ‘THU’ = Thursday • ‘FRI’ = Friday • ‘SAT’ = Saturday • ‘WKE’ = any weekend day • ‘WKD’ = any weekday • ‘ALL’ = any day, followed by the time of day (on a 24-hour clock), for example ‘WKD13:20’ is any weekday at 1:20 in the afternoon
ENDTIME	ending time for the speeds	specified like STARTTIME
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

4. Pocket Lane Table

Table 6 specifies the format for the pocket lane table. To validate a pocket lane table, it is necessary to verify the following:

- The field names and types are correct.
- The data values are in the legal ranges.
- The IDs are unique.
- The node and link references are correct.
- The lane number is that of a valid pocket lane.
- The offset and length are consistent with the setbacks and length of the link.
- None of the pockets overlap.
- All of the pocket lanes specified in the link table are present.

Table 6. Pocket lane table format.

Column Name	Description	Allowed Values
ID	ID number of the pocket lane	integer: 1 through 2,147,483,647
NODE	ID number of the node toward which the pocket lane leads	integer: 1 through 2,147,483,647
LINK	ID number of the link on which the pocket lane lies	integer: 1 through 2,147,483,647
OFFSET	starting position of the pocket lane, measured (in meters) from NODE (pull-out pockets only)	non-negative floating-point number
LANE	lane number of the pocket lane	integer: 1 through 255
STYLE	type of the pocket lane	one character: • ‘T’ = turn pocket • ‘P’ = pull-out pocket • ‘M’ = merge pocket
LENGTH	length of the pocket lane (in meters); turn pockets and merge pockets always start or end at the appropriate limit line	positive floating-point number
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

5. Lane Use Table

Table 7 specifies the format for the lane use table.

Table 7. Lane use table format.

Column Name	Description	Allowed Values
NODE	ID number of the node toward which the lane leads	integer: 1 through 2,147,483,647
LINK	ID number of the link on which the lane lies	integer: 1 through 2,147,483,647
LANE	lane number	integer: 1 through 255
VEHICLE	vehicle type(s) to which restriction applies	string of characters separated by slashes: • 'HOV2' = high occupancy vehicle with two or more occupants • 'HOV3' = high occupancy vehicle with three or more occupants • 'HOV4' = high occupancy vehicle with four or more occupants • 'BICYCLE' = bicycle • 'AUTO' = private auto • 'TRUCK' = motor carrier • 'BUS' = bus • 'TROLLEY' = trolley • 'STREETCAR' = streetcar • 'LIGHTRAIL = light rail transit • 'RAPIDRAIL' = rail rapid transit • 'REGIONRAIL' = regional rail
RESTRICT	type of lane restriction	one character: • 'O' = only this vehicle type may use lane • 'R' = lane required to be used by this vehicle type • 'N' = lane not allowed to be used by this vehicle type

Table 7 continued.

Column Name	Description	Allowed Values
STARTTIME	starting time for the restriction	a character string with the day of week, • 'SUN' = Sunday • 'MON' = Monday • 'TUE' = Tuesday • 'WED' = Wednesday • 'THU' = Thursday • 'FRI' = Friday • 'SAT' = Saturday • 'WKE' = any weekend day • 'WKD' = any weekday • 'ALL' = any day, followed by the time of day (on a 24-hour clock), for example 'WKD13:20' is any weekday at 1:20 in the afternoon
ENDTIME	ending time for the restriction	specified like STARTTIME
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

6. Parking Table

Table 8 specifies the format for the parking table.

Table 8. Parking table format.

Column Name	Description	Allowed Values
ID	ID number of the parking place	integer: 1 through 2,147,483,647
NODE	ID number of the node toward which vehicles are traveling	integer: 1 through 2,147,483,647
LINK	ID number of the link on which the parking place lies	integer: 1 through 2,147,483,647
OFFSET	location of the parking place, measured (in meters) from NODE	non-negative floating-point number
STYLE	type of the parking place	five characters: • 'PRSTR' = parallel on street • 'HISTR' = head in on street • 'DRVWY' = driveway • 'LOT' = parking lot • 'BNDRY' = network boundary
CAPACITY	number of vehicles the parking place can accommodate; zero for unlimited capacity	integer: 0 through 65,535
GENERIC	whether the parking place represents generic parking (not an actual driveway, lot, etc., but a group/aggregate of them used to simplify modeling)	one character: • 'T' = true/yes • 'F' = false/no
VEHICLE	type of vehicle(s) allowed to park at the parking place	string of characters separated by slashes: • 'AUTO' = private auto • 'TRUCK' = motor carrier • 'BICYCLE' = bicycle • 'TAXI' = paratransit • 'BUS' = bus • 'TROLLEY' = trolley • 'STREETCAR' = streetcar • 'LIGHTRAIL' = light rail transit • 'RAPIDRAIL' = rail rapid transit • 'REGIONRAIL' = regional rail • 'ANY' = any vehicle type

Table 8 continued.

Column Name	Description	Allowed Values
STARTTIME	starting time for parking	a character string with the day of week, • 'SUN' = Sunday • 'MON' = Monday • 'TUE' = Tuesday • 'WED' = Wednesday • 'THU' = Thursday • 'FRI' = Friday • 'SAT' = Saturday • 'WKE' = any weekend day • 'WKD' = any weekday • 'ALL' = any day, followed by the time of day (on a 24-hour clock), for example 'WKD13:20' is any weekday at 1:20 in the afternoon
ENDTIME	ending time for parking	specified like STARTTIME
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

7. Barrier Table

Table 9 specifies the format for the barrier table.

Table 9. Barrier table format.

Column Name	Description	Allowed Values
ID	ID number of the barrier	integer: 1 through 2,147,483,647
NODE	ID number of the node toward which vehicles are traveling	integer: 1 through 2,147,483,647
LINK	ID number of the link on which the barrier lies	integer: 1 through 2,147,483,647
OFFSET	starting position of the barrier, measured (in meters) from NODE	non-negative floating-point number
LANE	lane number of lane to the left of the barrier	integer: 0 through 255
STYLE	type of the barrier	ten characters: • 'CURB' = curb • 'BARRIER' = barrier • 'GRADESEP' = grade separation • 'STRIPE' = painted stripe • 'TEMPORARY' = temporary barrier
LENGTH	length of the barrier (in meters)	positive floating-point number
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

8. Transit Stop Table

Table 10 specifies the format for the transit stop table.

Table 10. Transit stop table format.

Column Name	Description	Allowed Values
ID	ID number of the stop	integer: 1 through 2,147,483,647
NAME	name of the stop	50 characters
NODE	ID number of the node toward which vehicles are traveling	integer: 1 through 2,147,483,647
LINK	ID number of the link on which the stop occurs	integer: 1 through 2,147,483,647
OFFSET	location of the stop, measured (in meters) from NODE	non-negative floating-point number
VEHICLE	types of vehicles for which this is a stop	string of characters separated by slashes: • 'BUS' = bus • 'TROLLEY' = trolley • 'STREETCAR' = streetcar • 'LIGHTRAIL' = light rail transit • 'RAPIDRAIL' = rail rapid transit • 'REGIONRAIL' = regional rail
STYLE	type of the stop	ten characters: • 'STOP' = stop (no station) • 'STATION' = station • 'YARD' = vehicle storage lot
CAPACITY	number of vehicles the stop can simultaneously handle; zero for unlimited capacity	integer: 0 through 65,535
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

9. Lane Connectivity Table

Table 11 specifies the format for the lane connectivity table. To validate a lane connectivity table, it is necessary to verify the following:

- The field names and types are correct.
- The data values are in the legal ranges.
- The node, link, and lane references are correct.
- Each lane has at least one incoming and at least one outgoing connection.

Table 11. Lane Connectivity table format.

Column Name	Description	Allowed Values
NODE	ID number of the node	integer: 1 through 2,147,483,647
INLINK	ID number of the incoming link	integer: 1 through 2,147,483,647
INLANE	lane number of the incoming lane	integer: 1 through 255
OUTLINK	ID number of the outgoing link	integer: 1 through 2,147,483,647
OUTLANE	lane number of the outgoing lane	integer: 1 through 255
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

10. Turn Prohibition Table

Table 12 specifies the format for the turn prohibition table.

Table 12. Turn prohibition table format.

Column Name	Description	Allowed Values
NODE	ID number of the node	integer: 1 through 2,147,483,647
INLINK	ID number of the incoming link	integer: 1 through 2,147,483,647
OUTLINK	ID number of the outgoing link	integer: 1 through 2,147,483,647
STARTTIME	starting time for the prohibition	a character string with the day of week, • 'SUN' = Sunday • 'MON' = Monday • 'TUE' = Tuesday • 'WED' = Wednesday • 'THU' = Thursday • 'FRI' = Friday • 'SAT' = Saturday • 'WKE' = any weekend day • 'WKD' = any weekday • 'ALL' = any day, followed by the time of day (on a 24-hour clock), for example 'WKD13:20' is any weekday at 1:20 in the afternoon
ENDTIME	ending time for the prohibition	specified like STARTTIME
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

11. Unsignalized Node Table

Table 13 specifies the format for the unsignalized node table. To validate an unsignalized node table, it is necessary to verify the following:

- The field names and types are correct.
- The data values are in the legal ranges.
- The node and link references are correct.
- Each incoming link entering an unsignalized node has a record.

Table 13. Unsignalized node table format.

Column Name	Description	Allowed Values
NODE	ID number of the node	integer: 1 through 2,147,483,647
INLINK	ID number of the incoming link	integer: 1 through 2,147,483,647
SIGN	type of sign control on the link	one character: • 'S' = stop • 'Y' = yield • 'N' = none
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

12. Signalized Node Table

Table 14 specifies the format for the signalized node table. To validate a signalized node table, it is necessary to verify the following:

- The field names and types are correct.
- The data values are in the legal ranges.
- The node references are correct.
- The plan references are correct.
- Each node has either one signalized or one unsignalized control.
- All plans are used.
- The start times are valid.

Table 14. Signalized node table format.

Column Name	Description	Allowed Values
NODE	ID number of the node	integer: 1 through 2,147,483,647
TYPE	type of the signal	one character: • 'T' = timed • 'A' = actuated
PLAN	ID number of a timing plan	integer: 1 through 65,535
OFFSET	relative offset (in seconds) for coordinated signals	non-negative floating-point number
STARTTIME	starting time for the plan	a character string with the day of week, • 'SUN' = Sunday • 'MON' = Monday • 'TUE' = Tuesday • 'WED' = Wednesday • 'THU' = Thursday • 'FRI' = Friday • 'SAT' = Saturday • 'WKE' = any weekend day • 'WKD' = any weekday • 'ALL' = any day, followed by the time of day (on a 24-hour clock), for example 'WKD13:20' is any weekday at 1:20 in the afternoon
COORDINATR	ID number of coordinator for the signal; equivalent to NODE ID number if signal is isolated	integer: 1 through 2,147,483,647
RING	single or dual ring, required only for TYPE = 'A'	one character: • 'S' = single • 'D' = dual
ENTRY	single or dual entry, required only for RING = 'D'	one character: • 'S' = single • 'D' = dual
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

13. Phasing Plan Table

Table 15 specifies the format for the phasing plan table. To validate a phasing plan table, it is necessary to verify the following:

- The field names and types are correct.
- The data values are in the legal ranges.
- The plan, phase, node, and link references are correct.

- Each incoming and outgoing link is controlled.

Table 15. Phasing plan table format.

Column Name	Description	Allowed Values
NODE	ID number of the node	integer: 1 through 2,147,483,647
PLAN	ID number of the timing plan	integer: 1 through 65,535
PHASE	phase number	integer: 1 through 255
INLINK	ID number of the incoming link	integer: 1 through 2,147,483,647
OUTLINK	ID number of the outgoing link	integer: 1 through 2,147,483,647
PROTECTION	movement protection indicator	one character: • 'P' = protected • 'U' = unprotected • 'S' = unprotected after stop
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

14. Timing Plan Table

Table 16 specifies the format for the timing plan table. To validate a timing plan table, it is necessary to verify the following:

- The field names and types are correct.
- The data values are in the legal ranges.
- The (plan, phase) pairs are unique.
- The time values are consistent.
- The phase sequence references existent phases.

Table 16. Timing plan table format.

Column Name	Description	Allowed Values
PLAN	ID number of a timing plan	integer: 1 through 65,535
PHASE	phase number	integer: 1 through 255
NEXTPHASES	phase number(s) of the next phase(s) in sequence	string of phase numbers, separated by slashes
GREENMIN	minimum length (in seconds) of the green interval, or fixed green length for timed signal	non-negative floating-point number
GREENMAX	maximum length (in seconds) of the green interval	non-negative floating-point number
GREENEXT	length (in seconds) of the green extension interval	non-negative floating-point number
YELLOW	length (in seconds) of the yellow interval	non-negative floating-point number
REDCLEAR	length (in seconds) of the red clearance interval	non-negative floating-point number
GROUPFIRST	for pre-timed or single ring: 1 if first phase, 0 if not first phase; for dual ring: number of phase group for which this phase is first phase, 0 if not first phase in the phase group	integer: 0 through 255
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

15. Detector Table

Table 17 specifies the format for the detector table.

Table 17. Detector table format.

Column Name	Description	Allowed Values
ID	ID number of the detector	integer: 1 through 2,147,483,647
NODE	ID number of the node toward which vehicles are traveling	integer: 1 through 2,147,483,647
LINK	ID number of the link on which the detector lies	integer: 1 through 2,147,483,647
OFFSET	starting position of the detector, measured (in meters) from NODE	non-negative floating-point number
LANEBEGIN	lane number of lane at which the detector begins	integer: 1 through 255
LANEEND	lane number of lane at which the detector ends, equal to LANEBEGIN for detector that lies on single lane	integer: 1 through 255
LENGTH	length of the detector (in meters)	non-negative floating-point number
STYLE	type of the detector	ten characters: • 'PRESENCE' = sense vehicles on detector • 'PASSAGE' = sense vehicles crossing detector
COORDINATR	ID number of coordinators interested in detector output	string of coordinator ids separated by slashes
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

16. Signal Coordinator Table

Table 18 specifies the format for the signal coordinator table.

Table 18. Signal coordinator table format.

Column Name	Description	Allowed Values
ID	ID number of the signal coordinator	integer: 1 through 2,147,483,647
TYPE	type of coordinator	ten characters: values to be determined
ALGORITHM	control algorithm used by coordinator	ten characters: values to be determined
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

17. Study Area Link Table

Table 19 specifies the format for the study area link table.

Table 19. Study area link table format.

Column Name	Description	Allowed Values
ID	ID number of the link	integer: 1 through 2,147,483,647
BUFFER	whether the link is in the buffer area or the study area	one character: • 'Y' = in buffer area • 'N' = in study area
NOTES	character string used for data quality annotations; free format (may be blank)	255 characters

C. Example Network Data Files

An example set of network data tables is available in files from the TRANSIMS package:

- TestNode.tbl: Test Node Table
- TestLink.tbl: Test Link Table
- TestPocket.tbl: Test Pocket Lane Table
- TestParking.tbl: Test Parking Table
- TestConnectivity.tbl: Test Lane Connectivity Table
- TestUnsignalized.tbl: Test Unsignalized Node Table
- TestSignalized.tbl: Test Signalized Node Table
- TestPhasing.tbl: Test Phasing Plan Table
- TestTiming.tbl: Test Timing Plan Table
- TestSpeed.tbl: Test Speed Table
- TestUse.tbl: Test Lane Use Table
- TestStop.tbl: Test Transit Stop Table
- TestCoordinator.tbl: Test Signal Coordinator Table
- TestDetector.tbl: Test Detector Table
- TestProhibition.tbl: Test Turn Prohibition Table
- TestBarrier.tbl: Test Barrier Table
- TestStudy.tbl: Test Study Area Link

(These tables are not reproduced in this document because of their length.) Figure 1 below shows the example network.

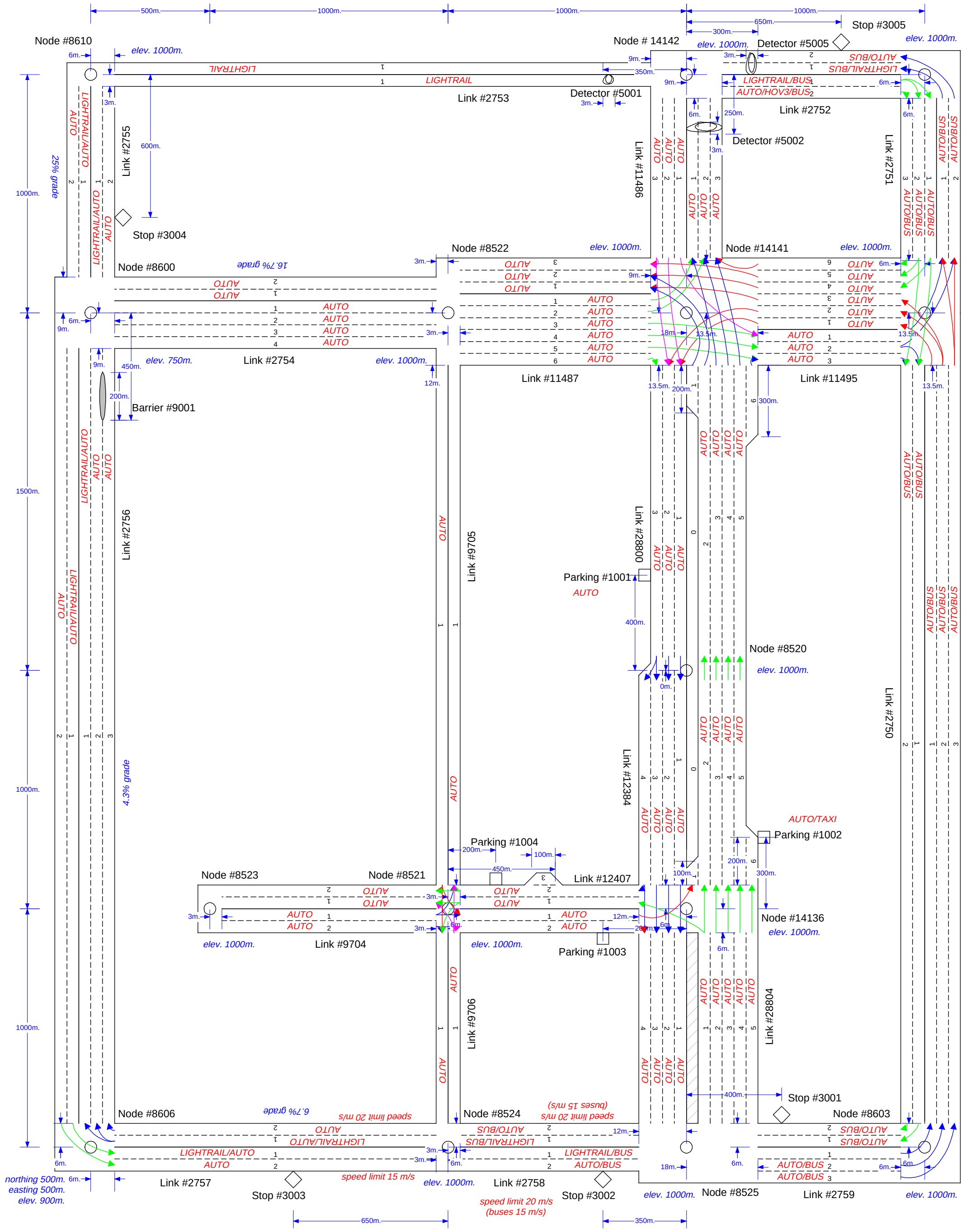

Figure 1. Example road network.

Note the following about this network:

- Three of the nodes (8521, 14136, and 14141) in the node are associated with intersections, while another (8520) is associated with a change in the permanent number of lanes.
- Entries in the link table illustrate how the permanent number of lanes, left pocket lanes, and right pocket lanes are counted and numbered.

- All three types of pocket lanes (turn pocket, merge pocket, and pull-out pocket) are represented in the pocket lane table.
- Several types of parking (lot, street, driveway, and generic vs. actual) are represented in the parking table.
- The first six rows in the lane connectivity table may be understood as follows: Lanes 1 and 2 on link 11487 (attached to node 14141) are exclusive left-turn lanes connecting only to lanes 1 and 2 on link 11486. Lanes 3, 4, and 5 on link 11487 are through lanes to lanes 1, 2, and 3 on link 11495. Lane 6 on link 11487 is a right-turn-only lane connecting to lane 3 on link 28800. Lane 3 on link 11486 connects to both lane 2 on link 28800 and lane 3 on link 11487, as shown in rows 9 and 10 of the table.
- The number of permanent lanes changes from three lanes on link 28800 to four lanes on link 12384 at node 8520. No right-of-way sign control is required at this node. A stop sign is indicated on link 12407 at node 14136, with no sign control on the other two links at this node. The unsignalized node table illustrates these.
- Node 14141 has a pre-timed signal control with an offset of 19.0 seconds. A single timing and phasing plan is always in effect. Node 8521 is defined as having an actuated signal and two timing and phasing plans. The signalized node table illustrates these.
- The movements permitted during phase 1 at node 14141 are through movements between links 11487 and 11495, as well as right-turn movements from these links. Additionally, unprotected right turns are permitted from links 11486 and 28800 during phase 1. The first six rows of the phasing plan table specify this information.
- Plan 1 in the timing plan table was specified in the signalized node table as applicable to node 14141. This is a timed signal with green, yellow, and red clearance intervals as indicated in row 1 of the table. Plans 2 and 3 for node 8521 were invented as illustrations and may not make sense as real timing plans.

D. Generating Missing Data

In this section we outline algorithms that one can use to fill in missing data in the network data tables if the data is otherwise unavailable or difficult to collection.

1. Generating "Through" Links

The "through" links referred to in Table 4 can be estimated using the following algorithm:

- Consider the links at a node for a given incoming link.
- The straight-through link is the link of the same functional class with the smallest deflection angle, provided the deflection angle is less than 60°.
- If there is no outgoing link of the same functional class with deflection angle less than 60°, then the straight-through link is the outgoing link with the smallest deflection angle.

2. Generating Link Colors

The link colors referred to in Table 4 can be estimated using the following algorithm:

- Start with an arbitrary node in the network and uniquely color all links connected directly to it.
- The uncolored links at the opposite ends of these links are then uniquely colored.
- This process is repeated until all links have been colored.

3. Generating Lane Connectivity

The TRANSIMS lane connectivity table (Table 11 above) provides information that indicates which outgoing lanes and links are valid for subsequent movements of vehicles on lanes entering an intersection.* The connectivity generation algorithm uses the information from the TRANSIMS node, link, and pocket lane tables (Table 3, Table 4, and Table 6 above) that describes the geometry of the network, the placement of permanent and pocket lanes, and the functional classes of links.

To create the lane connectivities, we examine intersections and determine each link's relative direction into the intersection, using the coordinates of the link's end points (nodes). Each incoming and outgoing lane of an intersection must have connectivity with at least one outgoing or incoming lane, respectively. To guarantee connectivity for all incoming lanes, the algorithm is written from the perspective of the normal traffic flow—i.e., in the normal traveling direction. To guarantee connectivity for all outgoing lanes, a comparable algorithm is written from the perspective of traffic flow in the reverse direction. To reuse some techniques of the first algorithm in the latter case, the network is inverted to reverse the incoming and outgoing concepts. The connectivities produced by the second algorithm must be reversed again, and then a union of the connectivities from the two algorithms is formed to complete the process.

a) Rotating the Intersection

For each link into the intersection, the lane connectivities for the incoming lanes (i.e., the lanes entering the intersection) are calculated by rotating the intersection (or comparably, by rotating the coordinate system) such that the *subject* link's incoming lanes are headed north as shown in Figure 2. The other links' incoming lane connectivities are not determined until they, in turn, are rotated into the north pointing position and become the *subject* link.

* The lane connectivity algorithm has been modified since this description was written. The description in this document gives the general flavor of how the lane connectivities are generated, but it does not quite correspond to the most recent version of the coded algorithm. When the updated documentation is complete, it will be made available.

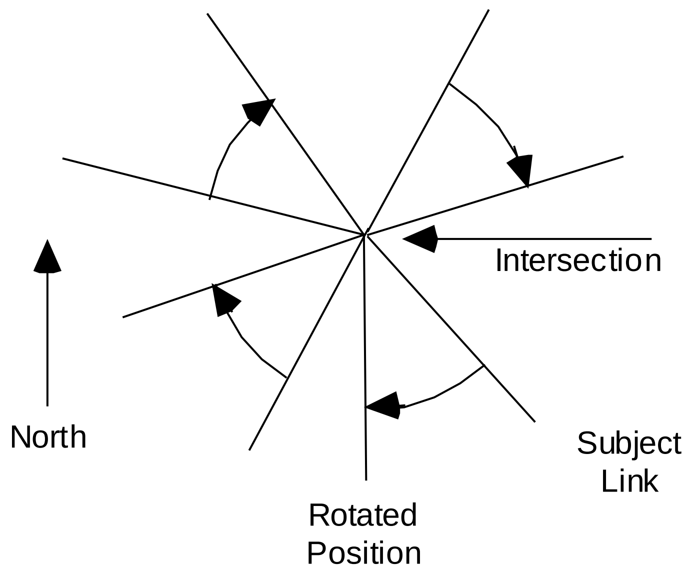

Figure 2. Intersection rotation with the subject link in the north pointing position.

Rotating the intersection is simply a transformation of coordinates with a slight modification. Because we rotate the subject link to 180°, the transformation equations differ in sign from the usual set,

$$x' = -x \cos(\Phi_s) + y \sin(\Phi_s),$$
$$y' = -y \cos(\Phi_s) - x \sin(\Phi_s), \textit{and}$$
$$\cos(\Phi') = \frac{y'}{L},$$

where Φ_S is the subject link's angle, *x* and *y* are the relative coordinates of the link's node that is not the intersection, and *L* is the subject link's length. One can compute the trigonometric functions for the subject link because we have the link's endpoint coordinates.

b) Lane Connectivity Algorithm

Considerable analysis of the intersection characteristics is necessary before the lane connectivities are created. The algorithm to determine the lane connectivities for the subject link proceeds in five steps:

1. Find the most likely through link.
2. Determine through lanes, left lanes, and right lanes.
3. Create through lane connectivities.
4. Create left lane connectivities.
5. Create right lane connectivities.

c) Finding the Most Likely Through Link

The search for the most likely through link (Figure 3) proceeds by examining each link's cosine, with a few caveats, and selecting the one with the maximum cosine. First, only those links with outgoing lanes and angles less than 90° (cosines greater than zero) are located. Next, since the most likely through link probably will have the same functional class as the subject link, our search initially is restricted to links with the identical functional class. On the other hand, if links with the same functional class, but not in the forward direction, exist, then links with different functional classes may be considered as possible through links. So, the initial search is limited to an angle relative to exactly straight ahead, say to 60°. If this initial search fails to find a most likely through link, then the functional class and 60° angle restrictions are ignored and a second search is performed. It is possible that a most likely through link will not be found, in which case all of the subject link's lanes will be designated turning lanes.

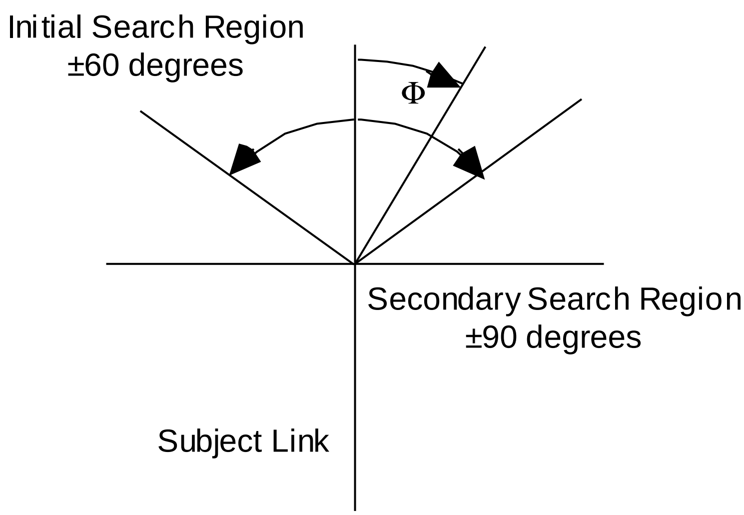

Figure 3. Search for the most likely through link.

An additional exception in determining the through link exists when the subject link is a freeway ramp. In general, the through link is not expected to be another ramp, though there may be a ramp back onto the freeway located on the other side of the intersection. To force the algorithm to search for other possible through links onto which to route the vehicles, other ramps are excluded from being the <u>through link</u> in such situations. However, the outgoing ramp link is considered as either a left link or right link in the algorithm and <u>connectivity</u> from the "off" ramp to the "on" ramp is allowed.

d) Determine Through Lanes, Left Lanes, and Right Lanes

To track which links are left and right of the through link, we establish the following comparison function:

$$f_\lambda = 2 - \cos(\Phi_\lambda) \qquad \text{if } x > 0,$$
$$f_\lambda = \cos(\Phi_\lambda) \qquad \text{if } x \le 0,$$

where Φ_λ is the angle for link λ and x is the ordinate of the link's other node relative to the intersection node. As shown in Figure 4, this function decreases monotonically counterclockwise from the subject link.

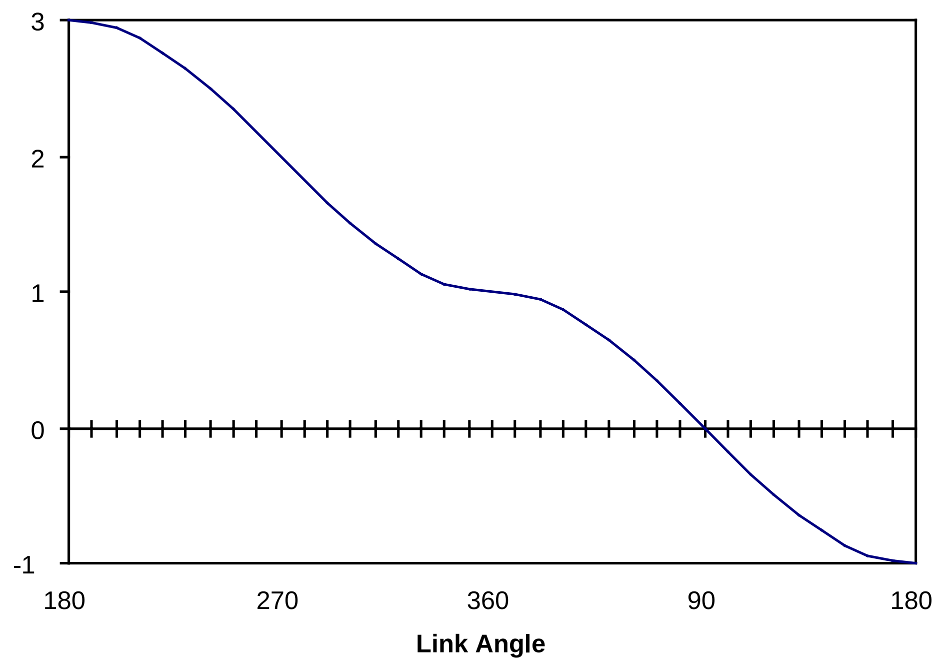

Figure 4. Link comparison function.

Thus, for right-bound links r, $f_r > f_t$, and for left-bound links l, $f_l < f_t$, where t designates the through link. If there is no through link, then $f_t = 1$.

(1) Through Lanes

The through-lane connectivities depend on the number of incoming permanent and turn pocket lanes on the subject link, the number of outgoing permanent and turn pocket lanes on the through link, and the presence of incoming and outgoing lanes, both left and right. The objective is to identify an equal number of incoming and outgoing through lanes on the subject and through links, respectively.

We are concerned about the presence of left outgoing lanes because if there are none, then the leftmost lane on the subject link is either a through lane, or possibly a right-turn lane if there also is no through link. Likewise, the presence of right outgoing lanes is cause for concern because their absence has connectivity implications for the rightmost lane. Thus, we make the following definitions:

- `LeftOutBound` is the number of outgoing permanent lanes on the left outgoing links, and
- `RightOutBound` is the number of outgoing permanent lanes on the right outgoing links.

Merge pockets are not considered in the current algorithm, so only the permanent outgoing lanes are counted.

Similarly, special attention must be given to the left and right incoming lanes, particularly if the through link has substantially more outgoing lanes than the subject link can feed as through lanes. The outgoing through lanes may be shifted left or right to accommodate the availability of the left and right incoming lanes that could turn onto the through link. (This algorithm does not account for prohibited left turns for which the connectivities have to be adjusted manually.) Again, we define:

- `LeftInBound` is the number of incoming lanes (including pocket lanes) on the left incoming links, and
- `RightInBound` is the number of incoming lanes (including pocket lanes) on the right incoming links.

We also define:

- `OutBound` is the number of outgoing permanent lanes on the through outgoing link,
- `InBound` is the number of permanent and turn pocket lanes on the subject link, and
- `LaneOffset` is the number of outgoing left-turn pocket lanes on the through link.

`LaneOffset` ensures that the through-link lane numbering accounts for the left-turn pocket lanes at the other through-link node (the through-link node away from the intersection for which we are building connectivities).

We initialize the limits of the outgoing through lanes,

$$\texttt{FirstTOL} = \texttt{LaneOffset} + 1,$$
$$\texttt{LastTOL} = \texttt{OutBound} + \texttt{FirstTOL} - 1,$$

where

- `FirstTOL` is the leftmost through outgoing lane on the through link, and
- `LastTOL` is the rightmost through outgoing lane on the through link.

Similarly, we determine the leftmost and rightmost incoming through lanes:

$$\begin{array}{ll} \texttt{FirstTIL} = \texttt{LPin} + 1 & \text{if } \texttt{LeftOutBound} > 0, \\ \texttt{FirstTIL} = 1 & \text{if } \texttt{LeftOutBound} = 0, \\ \texttt{LastTIL} = \texttt{LPin} + \texttt{LanesIn} & \text{if } \texttt{RightOutBound} > 0, \\ \texttt{LastTIL} = \texttt{LPin} + \texttt{LanesIn} + \texttt{RPin} & \text{if } \texttt{RightOutBound} = 0, \end{array}$$

where

- `FirstTIL` is the leftmost through incoming lane on the subject link,
- `LastTIL` is the rightmost through incoming lane on the subject link,
- `LPin` is the number of incoming left pocket lanes on the subject link,
- `LanesIn` is the number of incoming permanent lanes on the subject link, and
- `RPin` is the number of incoming right pocket lanes on the subject link.

Figure 5 shows a possible intersection configuration illustrating these definitions.

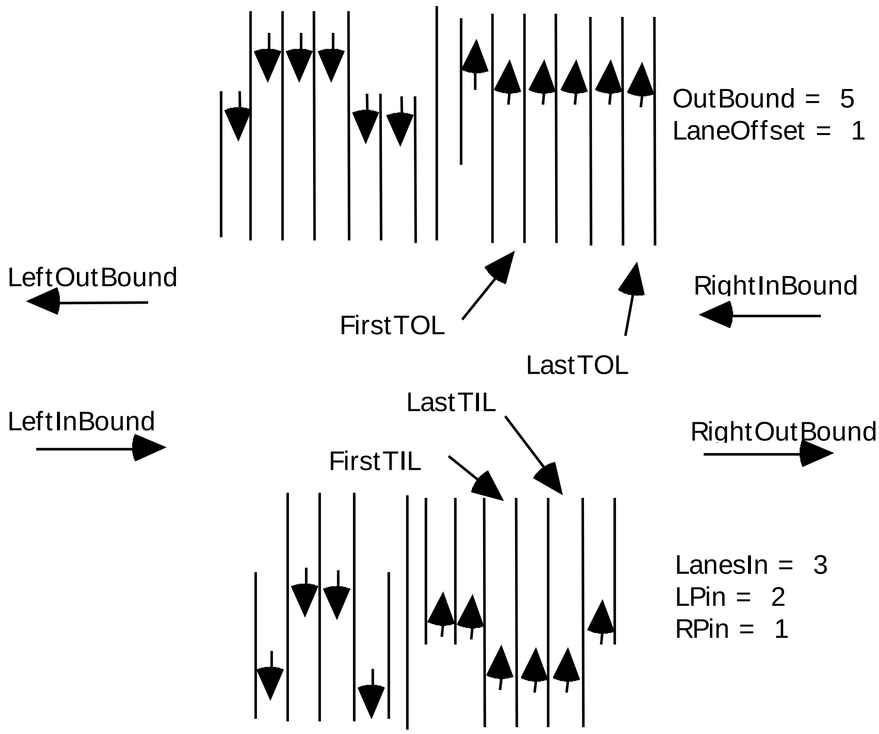

Figure 5. Initial determination of through lanes.

In this scheme the permanent lanes are the only through lanes unless pocket lanes are denied any connectivity as determined by the availability of outbound lanes on the left and right. That is, if there are no outbound lanes on the left, then any incoming left pocket lanes must be considered as through lanes. Similar considerations apply when outbound lanes are not available on the right.

The number of possible through outbound lanes and the number of possible through inbound lanes is calculated:

$$\texttt{ThruOutBound} = \texttt{LastTOL} - \texttt{FirstTOL} + 1, \text{ and}$$
$$\texttt{ThruInBound} = \texttt{LastTIL} - \texttt{FirstTIL} + 1.$$

To assure that the numbers of incoming and outgoing through lanes are equal, the smaller is chosen:

$$\texttt{ThruBound} = \min(\texttt{ThruOutBound}, \texttt{ThruInBound}).$$

Whether this leaves extra incoming lanes or extra outgoing lanes that will need to be changed from through lanes into turning lanes instead of through lanes is determined:

$$\texttt{ThruExcess} = \texttt{ThruInBound} - \texttt{ThruOutBound}.$$

If there are excess through lanes on the subject or through link, then the appropriate leftmost and rightmost through lane positions need to be adjusted to account for the excess. Also, the outbound right, left, and through lane availability must be calculated.

(2) Excess Incoming Through Lanes

If `ThruExcess` is greater than zero, there are more incoming through lanes than outgoing through lanes. Therefore, the excess lanes must be redistributed and reassigned as left and right-turn lanes for the subject link. The excess lanes are divided alternately between left and right turns, beginning with right turns. There may already be left- and right-turn lanes located on the subject link,

$$\texttt{LTurnPrev} = \texttt{FirstTIL} - 1,$$
$$\texttt{RTurnPrev} = \texttt{InBound} - \texttt{LastTIL},$$

where `InBound` is the total number of lanes entering the intersection from the subject link. This situation is illustrated in Figure 6.

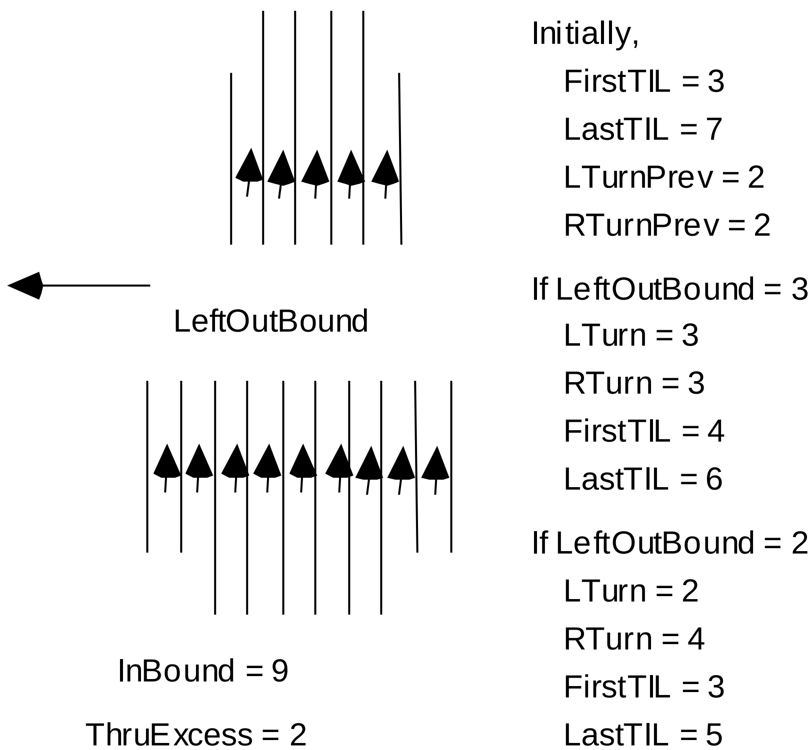

Figure 6. Example of excess incoming through lanes.

The left outbound lanes, totaling `LeftOutBound`, may not be able to accommodate all the left-turn lanes if half of the excess lanes are assigned as left-turn lanes. That is, half of the excess lanes are changed to left-turn lanes if, when added to `LTurnPrev`, the total is less than `LeftOutBound`; otherwise the total number of left-turn lanes is set to `LeftOutBound`,

$$\texttt{LTurn} = \texttt{LTurnPrev} + \text{floor}(\texttt{ThruExess}/2)$$
$$\text{if } \texttt{ThruExcess}/2 \leq (\texttt{LeftOutBound} - \texttt{LTurnPrev}),$$

$$\texttt{LTurn} = \texttt{LeftOutBound}$$
$$\text{if } \texttt{ThruExcess}/2 > (\texttt{LeftOutBound} - \texttt{LTurnPrev}),$$

where the *floor* function returns the largest integer smaller than or equal to the value. At this point, the number of right-turn lanes is calculated,

$$\texttt{RTurn} = \texttt{InBound} - \texttt{ThruBound} - \texttt{LTurn},$$

and the through-lane leftmost and rightmost positions are updated,

$$\texttt{FirstTIL} = \texttt{LTurn} + 1,$$
$$\texttt{LastTIL} = \texttt{InBound} - \texttt{RTurn}.$$

(3) Excess Outgoing Through Lanes

When `ThruExcess` is less than zero, there are more outgoing than incoming through lanes and these excess lanes must be reassigned as through-link outgoing lanes connected to incoming lanes from the left or right links. Note that we do not assign the left or right link incoming connectivities at this point; that is done when the left or right links are rotated to become the subject link. However, by apportioning the excess lanes to the left and right on the through link, the positions of the leftmost and rightmost through lanes on the through link are adjusted.

The assignment of excess lanes is biased to the right so that the right outgoing lanes can (possibly) accept right-turning vehicles without interference from through traffic. First, it must be determined whether there are sufficient lanes inbound from the left and right to service the apportioned excess lanes. The minimum between the number of left and right inbound lanes is found and used to determine whether that minimum can service half of the excess lanes. If not, the number of excess lanes assigned to that side of the through lanes is limited.

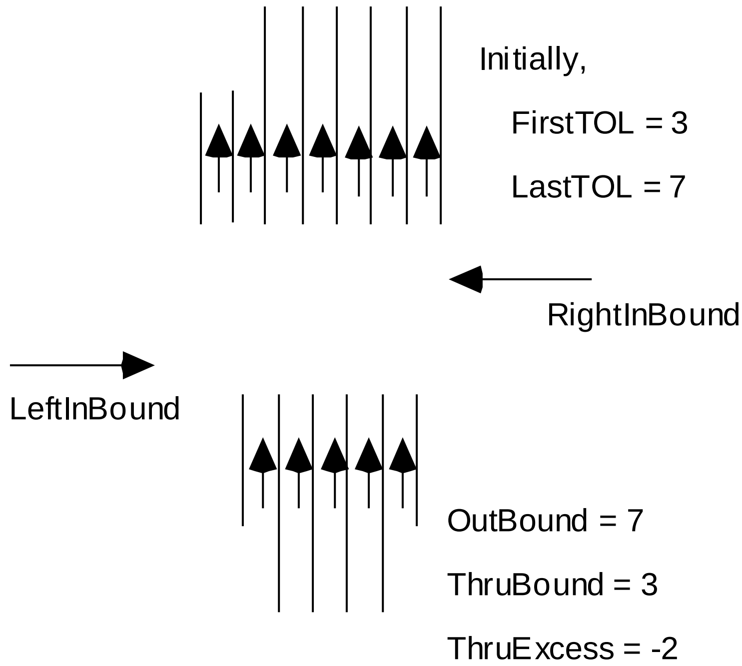

Figure 7. Example of excess outgoing through lanes.

Using Figure 7 as an illustrative example, when `LeftInBound` > `RightInBound` and `ThruExcess` > 2 (`RightInBound` – `OutBound` + `LastTOL`),

`LeftLanes` = `OutBound` – `ThruBound` – `RightInBound`,

or, when `LeftInBound` ≤ `RightInBound` and `ThruExcess` > 2 (`LeftInBound` – `FirstTOL` +1),

`LeftLanes` = `LeftInBound`;

otherwise,

`LeftLanes` = `FirstTOL` – 1 + floor(`ThruExcess`/2)

where `LeftLanes` is the total number of through-link lanes that are connected to lanes from the left links, and the *floor* function returns the largest integer smaller than or equal to the value.

Now, the leftmost and rightmost outgoing through lanes are redefined,

`FirstTOL` = `LeftLanes` + 1,
`LastTOL` = `LeftLanes` + `ThruBound`.

(4) Through-Lane Connectivities

At this point the lane connectivity table can be constructed. As shown in Figure 8, for the through lanes the incoming lane `FirstTIL` of the subject link is connected with the outgoing lane `FirstTOL` of the through link, {`FirstTIL` + 1} with {`FirstTOL` + 1}, etc., until incoming lane `LastTIL` is connected to outgoing lane `LastTOL`.

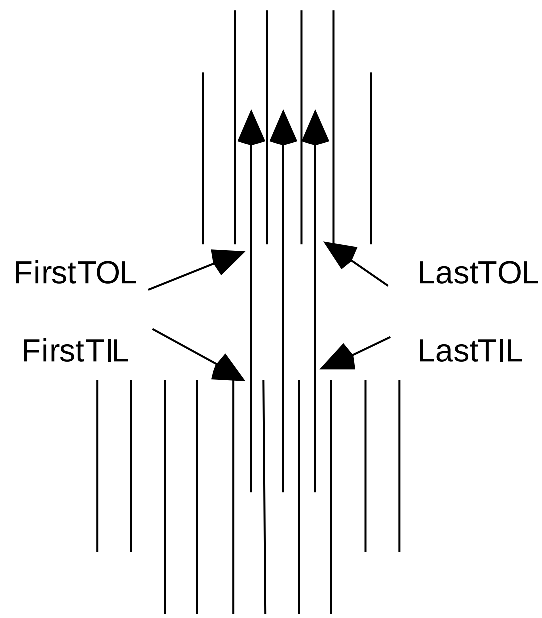

Figure 8. Example of through lane connectivities.

(5) Left-Turn-Lane Connectivities

Left-turn-lane connectivities are considered only if there are left outbound lanes. For the subject-link left-turn lanes, each left link is examined in a clockwise direction (the comparison function, f_λ, lets us order the links). First it is determined how many subject-link left-turn lanes must be assigned to the left link. All are assigned to this single left-turn link unless a second link exists, to which the remainder is assigned. As shown in Figure 9, we know the total outbound left lanes, `LeftOutBound`, and the total outbound lanes on this left link,

`LOut` = `LPOut` + `LanesOut` + `RPOut`,

where `LPOut`, `LanesOut`, and `RPOut` are defined for this left link using the same procedure as outlined earlier for determining through lanes. The difference between `LeftOutBound` and `LOut` is the number of lanes on additional left links available for left-turns from the subject link. So, from the `LTurn` left lanes, at least (`LTurn` – (`LeftOutBound` – `LOut`)) must be used on this left link. Also, any outgoing left pocket merge lanes must be found since they will be utilized first. So,

`OutLane` = `LPOut`
    if `LPOut` >= (`LTurn` – (`LeftOutBound` – `LOut`)),
`OutLane` = `LTurn` – (`LeftOutBound` – `LOut`),
    if `LPOut` < (`LTurn` – (`LeftOutBound` – `LOut`)),

though,

`OutLane` = 1,   if the previous result is less than or equal zero, where `OutLane` is the number of lanes on this left link to be connected to left turns from the subject link. Now each subject link's left-turn lane, starting at lane number one, is connected to each left link's lane, also starting at lane number one, incrementing up to `OutLane`.

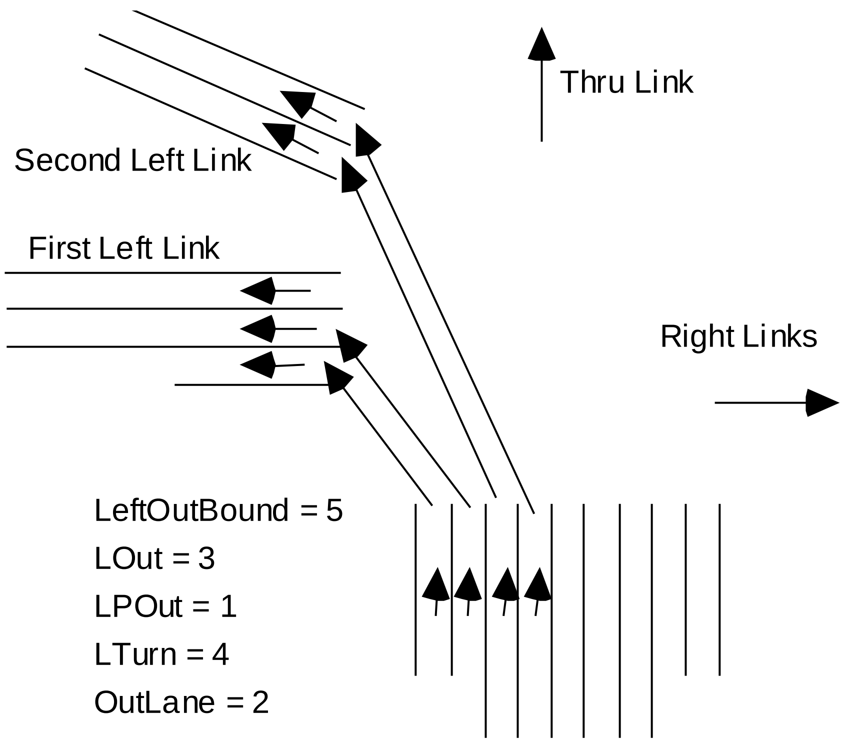

Figure 9. Example of left-turn-lane connectivities.

If there is another left link, the same procedure is applied, although `LeftOutBound` must be updated to reflect that the lanes of previous left links are no longer available, and `LTurn` must be reduced by `OutLane` because those lanes have already been utilized. Each subject link's left-turn lane must be connected (starting, not from 1, but from `OutLane` + 1, the next lane over from where we stopped on the previous left link). However, the subject link's left-turn lane cannot to exceed `FirstTIL`, the first through lane. This is avoided by always starting from lane number one.

This process is continued for each left link. If all the subject link's left-turn lanes have been used, and there are additional left links with outgoing lanes, the left turns from the subject link's first through lane, `FirstTIL`, are assigned to the first lane, lane number one, of the additional left links.

(6) Right-Turn-Lane Connectivities

As illustrated in Figure 10, the right-turn-lane connectivities are assigned in the same manner previously described for the left-turn lanes; we continue to scan the links in a clockwise direction and to assign the subject link's lanes from left to right. As a result, after all the right-turn lanes have been accounted for, any necessary additional right-turns from the subject link's far right lane are created. As before, the total outbound right lanes, `RightOutBound`, and the total outbound lanes on this right link,

`ROut` = `LPOut` + `LanesOut` + `RPOut`,

are known.

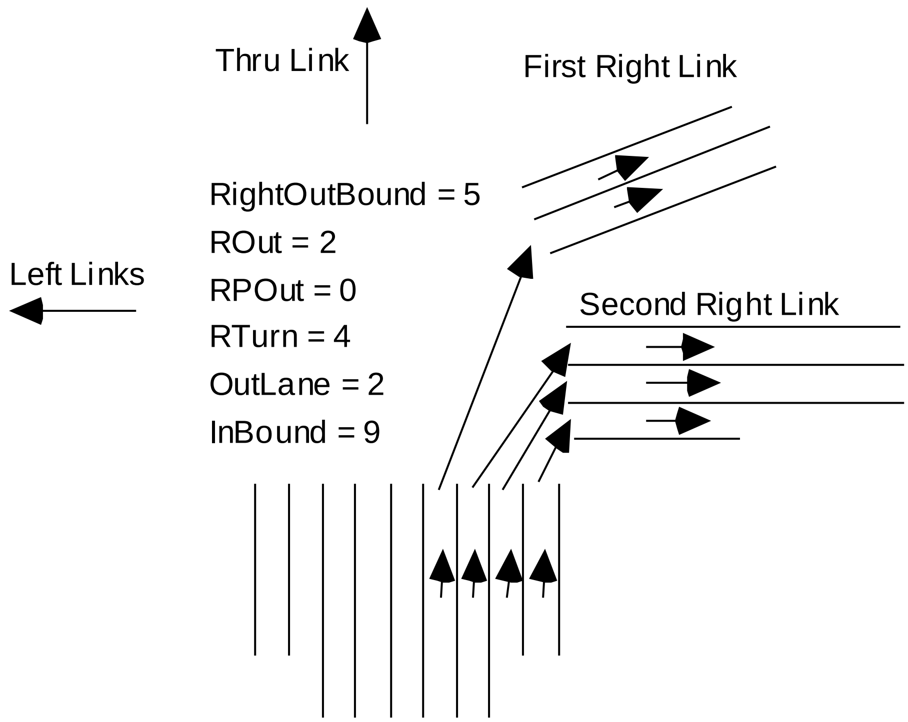

Figure 10. Example of right-turn-lane connectivities.

Using the same arguments outlined for the left-turn-lane connectivities, but applying them to the right lanes, yields

`OutLane` = `ROut` – `RPOut` + 1,
if `RPOut` >= (`RTurn` – (`RightOutBound` – `ROut`)),
`OutLane` = `RightOutBound` – `RTurn` +1,
if `RPOut` < (`RTurn` – (`RightOutBound` – `ROut`)),

though

`OutLane` = `ROut`, if the previous result is greater than `ROut`.

Beginning with lane {`InBound` – `RTurn` + 1} on the subject link and lane `OutLane` on the right link, the connectivity between the lanes, incrementing up to `Rout`, is assigned. For each additional link, the same procedure is followed, updating `RightOutBound` and `RTurn`. The process is continued for each right link. If all of the subject link's right-turn lanes have been utilized and there are additional right links with outgoing lanes, the right turns from the subject link's last lane, `InBound`, are assigned to the last lane, `ROut`, of the additional right links.

At this point the lane connectivities for the subject link are complete, and the intersection is rotated to select another subject link; or if the intersection is complete, the next intersection is selected and this process is repeated.

4. Generating Intersection Signs

To determine the sign controls at unsignalized intersections, the following ranking of functional classes must first be considered:

1. ‘FREEWAY’ = freeway
2. ‘XPRESSWAY’ = expressway
3. ‘PRIARTER’ = primary arterial
4. ‘FRONTAGE’ = frontage road
5. ‘SECARTER’ = secondary arterial
6. ‘COLLECTOR’ = collector
7. ‘LOCAL’ = local street
8. ‘OTHER’ = other
9. ‘RAMP’ = freeway ramp

Now apply the following rules to determine where to place traffic control signs [Ce 96]:

- Any intersection approach link of Class 1 or 2 should have no control sign.
- Any intersection approach link of Class 3 or 4 should have a stop control if all other approach links are Class 3 or 4, and no control sign if other links are higher-numbered classes.
- Any intersection approach link of Class 5 or 6 should have a stop control if all other approach links are of lower-numbered classes, and no control sign if other links are of higher-numbered classes.
- Any intersection approach link of Class 7 should have a stop control.
- Any intersection approach link of Class 9 should have a yield control.
- Any intersection approach link of Class 4 should have no control sign if the intersection is a “T,” and a stop control if the intersection is “four-legged.”

E. Software Tools

A number of network-related software tools are presently available for use in data preparation:

- *PrepareNetwork:* A command-line application program that automatically calculates “through” links and link “colors” for a network.
- *ValidateNetwork:* A command-line application program that checks the validity of network data tables.
- *ReadNetwork:* A command-line program that constructs C++ network objects in memory from a set of network data tables.
- *ManageDatabase and jManageDatabase:* command-line and graphical-user-interface application programs for managing (importing, exporting, viewing) data in the TRANSIMS database.
- *IOC-1 Lane Connectivity Generator*: An Oracle-based PL/SQL application for generating lane connectivity data tables.

- *IOC-1 Input Editor:* An ArcView-based graphical-user-interface application program for viewing and validating network data tables.

III. Population Data

The TRANSIMS synthetic population system is designed to produce populations (family households, non-family households, and group quarters) that are statistically equivalent to actual populations when compared at the level of block group or higher. The methodology used by this system is described in Reference [BBM 96]. The inputs to the software are U.S. Census Bureau data (STF3A and PUMS) and MABLE/GEOCORR data. Census Bureau STF3A and PUMS data formats are commonly used and are available on CD-ROM from the Census Bureau—these data inputs will not be described in any detail in this document. MABLE/GEOCORR data is relatively new, and is described in Reference [Bl].

References

[BBM 96] R. J. Beckman, K. A. Baggerley, and M. D. McKay, *Creating Synthetic Baseline Populations*, Statistics Group, Los Alamos National Laboratory, Los Alamos, NM, 1996.

[Bl] J. Blodgett, “MABLE/GEOCORR Geographic Correspondence Engine,” http://www.oseda.missouri.edu/plue/geocorr/doc/article.html.

[Ce 96] K. Cervenka, personal communication, 1996.

[Do 97] R. Donnelly, personal communication, 1997.

[GHA 88] Federal Highway Administration, *Manual on Uniform Traffic Control Devices*, (Washington, D.C.: U.S. Government Printing Office, 1988).

[ITE 85] Institute of Transportation Engineers, *Traffic Control Systems Handbook*, (Washington, D.C.: ITE Publications, 1985).

[ITE] Institute of Transportation Engineers, *Traffic Detector Handbook*, (Washington, D.C.: ITE Publications, n.d.).

[MM 84] M. D. Meyer and E. J. Miller, *Urban Transportation Planning*, (New York: McGraw-Hill, 1984).

[Or 93] F. L. Orcutt, Jr., *The Traffic Signal Book*, (Englewood Cliffs, New Jersey: Prentice Hall, 1993).

[PP 93] C. S. Papacostas and P. D. Prevedouros, *Transportation Engineering and Planning*, (Englewood Cliffs, New Jersey: Prentice Hall, 1993).